The Knowledge Engineering Review, Vol. 30:2, 187–200.
doi:10.1017/S0269888914000290

Adopting trust and assurance as indicators for the reassignment of responsibilities in multi-agent systems

BENJAMIN GÂTEAU[1], MOUSSA OUEDRAOGO[1], CHRISTOPHE FELTUS[1], GUY GUEMKAM[1], GRÉGOIRE DANOY[2], MARCIN SEREDYNSKI[3], SAMEE U. KHAN[4], DJAMEL KHADRAOUI[1] and PASCAL BOUVRY[2]

[1]*Public Research Centre Henri Tudor, 29 Avenue John F. Kennedy, L-1855 Luxembourg, Luxembourg;*
e-mail: benjamin.gateu@tudor.lu;
[2]*CSC Research Unit, University of Luxembourg, 6 rue Coudenhove Kalergi, L-1359 Luxembourg, Luxembourg;*
e-mail: gregoire.danoy@uni.lu, pascal.bouvry@uni.lu;
[3]*Interdisciplinary Centre for Security, Reliability and Trust, University of Luxembourg, 6 rue Coudenhove Kalergi, L-1359 Luxembourg, Luxembourg;*
e-mail: marcin.seredynski@uni.lu;
[4]*NDSU-CIIT Green Computing and Communications Laboratory, Department of Electrical and Computer Engineering, North Dakota State University, Fargo, ND 58108-6050, USA;*
e-mail: samee.khan@ndsu.edu

Abstract

Multi-agent systems have been widely used in the literature, including for the monitoring of distributed systems. However, one of the unresolved issues in this technology remains in the reassignment of the responsibilities of monitoring agents when some of them become unable to meet their obligations. This paper proposes a new approach for solving this problem based on (a) the gathering of evidence on whether the agent can or cannot fulfil the tasks it has been assigned and (b) the reassignment of the task to alternative agents using their trust level as a selection parameter. A weather station case study is proposed as an instantiation of the proposed model.

1 Introduction

The adoption of multi-agent systems (MAS) to monitor highly distributed systems has been gaining momentum in the recent years. Such systems have witnessed crucial demand for deployment in diverse application scenarios, such as e-commerce, e-health, network intrusion detection, telematics and transport systems, and environmental monitoring (Baig, 2012). The rationale for the aforementioned systems mainly relies on the inherent properties and characteristics that the multi-agent technology offers. An agent is commonly considered as an encapsulated computer system that is situated in some environment and is capable of flexible and autonomous action within the environment to meet the design objectives (Wooldridge, 2002). As agents control their own behaviour, they may (and in many cases must) cooperate and negotiate with each other in order to achieve the desired goals (Jennings, 1999). The convergence of these agents' properties and distributed systems behaviour makes the multi-agent architecture an appropriate mechanism for the monitoring of network infrastructure, including the security aspects (Wooldridge, 2002).

Efficiently monitoring the network using an MAS imposes that agent meets its assigned objectives that we will hereafter refer to as responsibilities. In addition, the environment in which the system or network operates can be subject to unpredictable changes. Some agents may then become unable to meet their assigned responsibilities. For instance, hazards or even malicious actions could break communication

links between monitoring agents. Alternatively, the agents entrusted with conducting the verifications and measurements may fail to fulfil their responsibilities owing to (a) erroneous assignment of their rights or alteration of the latter during runtime, (b) insufficient agents' capabilities to accomplish all their tasks, and (c) an agent overload resulting in a failure to meet some of its tasks. Consequently, a solution where MASs are reinforced with the ability to reassign the responsibilities of faulty agents to others that provide similar capabilities is needed. This necessitates to first gather information on the aptitude or inaptitude of a given agent to meet its responsibilities. Such information can then form the basis for a trusted decision as to whether alternative agents should be sought for the fulfilment of a given task.

Despite a large body of work on the usage of MAS for networks and systems monitoring (Kolaczek & Juszczyszyn, 2007; Abielmona *et al.*, 2011), the assignment of the agents' functions is always undertaken before the MAS system deployment. Moreover, the reassignment of responsibilities to other agents in the event of an agent failure to conduct its tasks is impossible. Such a faulty agent may result in the monitoring system being grounded or induce the system to issue monitoring values that are either erroneous or incomplete. Needless to state that the gravity of the consequences of such a failure is stringently related to the criticality of the infrastructure or the system being monitored. Consequently, the core question addressed in this paper can be stated as: 'How to ensure the continuity of the MAS monitoring activities in a way that the failure of some of the agents to fulfil the corresponding responsibilities does not drastically affect system operations?'. The approach presented in this paper relies on the evaluation of the assurance that a given agent has the necessary credentials to fulfil the corresponding responsibility. A trust measure is used to trigger the automatic reassignment of responsibilities to a different agent in case of the occurrence of a hazard that results in one agent being isolated or unable to react promptly to a monitoring request. As a solution, we use a normative organisation modelling languages (OML) for MAS named *M*OISE^{Inst} (Gâteau *et al.*, 2005), which is an extension of *M*OISE^{+}[1] (Hubner *et al.*, 2002), to represent security policies provided by the responsibility model of Feltus and Petit (2009) as norms supervise their respect by agents. Information linked to the specification of norms is used as metrics to assure achievement. Respect of norms histories are used as input for the centralised evaluation of agent's reputation.

The remainder of this article is organised as follows. Section 2 discusses the related work in terms of OML, responsibility models in MAS, and monitoring of security. Section 3 presents the proposed responsibility model and the condition under which an agent is expected to meet the corresponding responsibility. In Section 4, we describe the reference case study adopted for the validation of our model. Section 5 presents an instantiation of the responsibility model through the assignment and the reassignment of the responsibilities. Finally, Section 6 provides our conclusions and perspectives.

2 Related work

2.1 Organisation modelling languages

Different models are manipulated within an MAS that can be described by using vowel approach AEIO (Agent, Environment, Interaction, Organisation) introduced in Demazeau (1995). *Agents* deal with the models (or architectures) used for the active part of the agent, from a simple automata to a complex knowledge-based system. *Environments* are the places where the agents are located. *Interactions* concern the infrastructures, the languages, and the interaction protocols between agents, from simple physical interactions to complex communicative acts. *Organisations* structure the agents in groups, hierarchies, and relations.

Our primary focus will be on the latter. That is to say that, how MAS organise agents and set up a self-governing behaviour through cooperation between agents? There exists two possible ways of obtaining an organisation in an MAS, namely: a bottom-up or a top-down approach. Contrary to the top-down approach, the bottom-up one does not manipulate any organisational model defined *a priori* but the organisation emerges from interactions between goal-oriented agents (with objectives dedicated to themselves and not to the global society). Indeed, this approach utilises some interaction capabilities to

[1] *M*OISE^{+}: Model of Organisation for MAS.

dynamically create and adapt the MAS organisations. In this work, we consider a whole hierarchical organisation of agents dedicated to infrastructure monitoring. The mapping of the MAS and architecture to the monitored system infrastructure justifies the choice of a top-down approach. To represent the complex social organisation within an MAS, OML have been proposed that rely on several dimensions, such as structural, functional, dialogic, environmental, and contextual.

The *structural* dimension represents the structure of the collective level of an MAS, generally in terms of roles/groups/links. Such a structural specification (SS) is used in AGR (Ferber & Gutknecht, 1998), Islander (Esteva *et al*., 2002), *M*OISE^{+}, and *M*OISE^{Inst}. The *functional* dimension specifies the global functioning of the system, as used in TAEMS (Lesser *et al*., 2004), TEAMCORE (Pynadath & Tambe, 2003), or *M*OISE^{+}. Some models, such as ISLANDER, add a *dialogical* dimension that specifies MAS interactions in terms of communications between agents. The *environmental* dimension allows to constrain the anchoring of the organisation in an environment, such as in AGRE (Ferber *et al*., 2004). Inspired by ISLANDER, *M*OISE^{Inst} introduced a *contextual* specification (CS) to define *a priori* the transition between different configurations of norms, structures, and plans (Boissier & Gâteau, 2007). We do not intend to provide an exhaustive comparison of the aforementioned OML in terms of the primitives or modelling power that each can offer. The interested readers are encouraged to refer to Coutinho *et al*. (2007) for a systematic comparison of OML.

As mentioned in Boissier and Gâteau (2007), depending on the various dimensions, the influence on the agents' behaviour may be quite different. In models, such as TAEMS, where only the functional dimension is specified, the organisation has nothing to 'tell' to the agents when no plan or task can be performed. Otherwise, if only the structural dimension is specified as in AGR, the agents have to reason for a global plan every time they want to work collectively. Because the agent's options are restricted by the structure, even with a smaller search space of possible plans, the problem is deemed a difficult proposition. Moreover, plans will be lost as no organisational memory exists. Therefore, in the context of open systems, we hypothesise that if the organisation model specifies both dimensions as in *M*OISE^{Inst} or TEAMCORE or a third one as in ISLANDER then the MAS that follows such a model can be more effective in leading the group behaviour to the desired objectives. On the agent's side, the models can develop richer reasoning abilities about agents and the organisation. Agents may gain more information on the possible cooperation (in terms of roles, groups, and on the possible goals, or on the performative structures) that may be conducted with other agents within the MAS.

Besides the aforementioned dimensions, the *deontic* and *normative* dimensions proposed in *M*OISE^{+} and ISLANDER or *M*OISE^{Inst}, respectively, address the agents autonomy problematic and consider organisations as normative constructs aiming at explicitly controlling the underlying MAS. While in other OMLs, the agents are supposed to be benevolent and must comply with the organisational specification (OS), the *M*OISE^{Inst} paradigm adds the possibility for agents to develop explicit reasoning on their autonomy with respect to the organisational constraints (Boissier & Gâteau, 2007).

2.2 Responsibility in multi-agent systems

A number of works have focused on the responsibility held by agents in an MAS system. According to Sommerville (2007), such a responsibility can be considered as a duty, held by some agents, to achieve, maintain, or avoid a given state, subject to conformance with organisational, social, and cultural norms. In this work, the responsibility is modelled using (a) the rights necessary for the agent to achieve a task or obligation, (b) the accountability, and (c) the assignment of tasks to agents that focuses on the necessity to have an agent commitment before the assignment. Li and Hoang (2009) have stressed the importance of using the responsibilities of a role in an organisation to dynamically interact with the agents but does not address the dynamic assignment of tasks to those agents. They proposed a role-interaction-organisation security model and applied the model to an e-health system, which is modelled as an MAS. These roles in this model not only determine access rights passively, but also initiate requests to interact dynamically with the agents who meet the security requirements. That is to say that, the confidential e-health data from unauthorised access is mandated. The interaction and the organisation models help in identifying the actions and responsibilities that a role can assume in the system within the organisation and any dynamic

interactions that the system can partake. In Guemkam *et al*. (2011), the authors have proposed an agent-based framework to support an alert mechanism for power distribution systems by using a reputation-based trust approach. The architecture provides a framework for dynamically assigning responsibilities to agents depending on the context (e.g. crisis situation). This permits the management of the agents' access rights towards critical information. This reputation-based trust measure relies on the similarity views between two agents during the assignment process. However, this scheme does not take into account the assurance or body for evidence that is necessary for an agent to fulfil a task.

2.3 Monitoring of security

MAS have also been extensively used for monitoring the security of a given system. For instance, the authors in Boudaoud *et al*. (2000) proposed an intelligent multi-agent approach for the design of an intrusion detection system (IDS) with a clear specification of the responsibilities of each monitoring agent. The proposed scheme operates at two layers. First, at a higher level, the manager layer operates and manages the security of the network. Three different agent types are involved. The *Security Policy Manager* agent manages the policies specified by the network security administrator. The *Intranet Manager* agents control the local agents that monitor the network traffic flow and report to them. These are managed by the *Extranet Manager* agent which assigns and delegates them intrusion detection tasks. Finally, the operations of the Extranet Manager agents are controlled by the policies of the Security Policy Manager agent.

Later, Kolaczek and Juszczyszyn (2007) proposed an attack pattern ontology and a formal framework within a distributed multi-agent IDS. In this approach it is assumed that the network system consists of a set of nodes. Two types of agents are considered. *Monitoring agents* observe the nodes, process captured information, and draw conclusions that are necessary to evaluate the current state of system security within their areas of responsibility. *Managing agents* are responsible for gathering information from *Monitoring agents* and generating reports about global threats and ongoing attacks. Each *Monitoring agent* monitors its area of responsibility that may consist of a set of network nodes.

Servin and Kudenko (2008) introduced a reinforced learning-based approach for the design of an intelligent multi-agent IDS to detect new and complex distributed attacks. In this reinforced learning architecture, each network sensor agent learns to interpret local state observations, and then communicates the information to a central agent higher up in the hierarchy. These central agents, in turn, learn to send signals up the hierarchy, based on the signals that they receive. The agent at the top of the hierarchy learns when to signal an intrusion alarm.

Abielmona *et al.* (2011) propose an architecture which is referred to as retroactive. It cumulates both proactive and reactive features and enables a given agent to implement a reaction strategy based on the occurrence of an event in the environment. By ensuring that each agent records the strategy associated with a given event, the architecture allows an autonomous robot to learn over a period of time. This ultimately prepares these robots to respond in real time to actual events in the environment, as they occur (Baig, 2012).

The BUGYO methodology (Ouedraogo *et al*., 2008) adopts a three-layer hierarchical MAS architecture for the monitoring of security assurance. The first level includes a single agent that is embedded within the server. When the server receives a request to perform a security assurance evaluation, the *server agent* handles the request and identifies the appropriate multiplexer agent or *MUX agent* (using a role directory). Finally, *probe agents* trigger the associated probe in the event of receiving a measurement request from an MUX agent. The probe agents also collect measurements from instrumentations and transmit them to the MUX agents. Similarly, Pham *et al.* (2008) present a security assurance evaluation approach based on attack graphs. The MAS architecture proposed in this approach is also built upon the BUGYO methodology. The concept of an attackability metric is introduced to characterise the possibility of attack along with other metrics for anomaly detection that assess both the static and dynamic visions of the underlying system.

In summary, the multi-agent-based architectures for the monitoring of distributed systems examine the capacity of a software agent to achieve tasks through respect of norms or rules, and as mechanism to ensure security. However, the aforementioned models do not propose any method to reassign the responsibilities to peers when a given agent fails to fulfil its task.

3 The responsibility model

We addressed several definition of responsibility in previous sections. In this work a responsibility is considered as a state assigned to an agent to denote the (a) obligations concerning a task, (b) accountabilities regarding its obligations, and (c) rights and capabilities necessary to perform tasks (Feltus & Petit, 2009). Main elements of our model are represented in green in Figure 2.

3.1 Responsibility and role

In our responsibility model, an obligation is a duty which links a responsibility with a task that must be performed. In *M*OISE^{Inst} a norms is an obligation or a permission that links a role with a mission to achieve.

More precisely, the accountability concept denotes a duty to justify achievement, maintenance, or avoidance of some given state to an authority under threat of sanction (Stahl, 2006). Therefore, the accountability parameter contributes to the valuation of trust. An agent's right encompasses facilities required by an agent to fulfil the desired obligations, such as the access right that the agent gets once it is assigned with a responsibility. Capability describes the required qualities, skills, or resources necessary to perform a task. The capability parameter depends on number of parameters relating the agents. These include the (a) ability to make decisions, (b) processing time, and (c) ability to analyse a problem and the location within the network. The commitment pledged by the agent represents the engagement to fulfil a task. The commitment concept has been the subject of many researches in MAS as explained in great detail in Singh (2008).

*M*OISE^{Inst} (Gâteau *et al*., 2007) is a language to describe normative organisational model through the definition of rights and duties of autonomous agents by means of unambiguous specifications. This language is founded on the *M*OISE^{+} organisational model (Hübner *et al*., 2002). It is composed of the following components that are used to specify an organisation of agents in terms of structure, functioning, evolution, and norms as depicted on Figure 1 (for a complete formal description of *M*OISE^{Inst} and organisation example see Gâteau, 2007). An OS is composed of:

- An SS defines (i) the *roles* that agents will play in the organisation, (ii) the *relations* between these roles in terms of authority, communication, or accointance, (iii) the *groups*, additional structural primitives used to define and organise sets of roles.

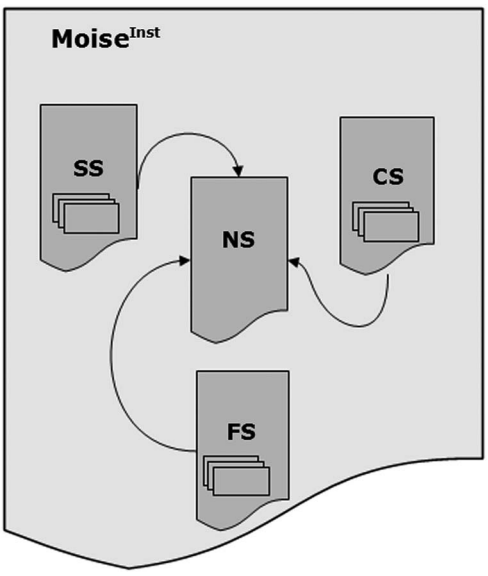

Figure 1 *M*OISE^{Inst}: a normative organisation specification language

- A *functional specification* (FS) defines global *business processes* that can be executed by the different agents participating to the organisation according to their roles and groups.
- A CS specifies, *a priori*, the possible evolution of the organisation in terms of *contexts* and *transitions* between the specifications (such as state/transition graph).
- A *normative specification* (NS) defines the deontic relations gluing the three independent specification (SS, FS, CS). Norms define rights (i.e. *permission*) or duties (i.e. *obligation*, *prohibition*) for a role or a group to execute a *mission* in a particular *context* and during a given *time*. This is supervised by an *issuer* which can apply a *sanction* on the *bearer* if the norm is not respected. A norm is active when the *context* referred in the norm equals the current organisation context. A norm is valid as long as its *condition* is satisfied. A norm could be respected or violated as long as it is active and valid. We represent a norm as the following expression:

$$norm : \varphi \rightarrow op(cont, issuer, bearer, m, sanc, w, tc)$$

where φ is the condition that defines the particular state of the organisation in which the norm may be valid; $op \in \{O, P, F\}$ defines if the norm is an obligation (O), a permission (P), or a prohibition (F); *cont* the context of the CS in which the norm becomes active (see below); *issuer* and *bearer* refer to structural entities of the SS (i.e. the whole groups and roles) from which the norm is issued and on which it is applied. Let us notice that the expression of norms refers to the notions of roles and groups and not to agents themselves. In this way, the norm expressions are independent of the kinds of agents that could populate the system at one time. *m* the mission of the FS concerned by the norm; *sanc* contains the reference of a different norm in the NS. It expresses a 'sanction' to apply in case of norm violation. *w* defines a priority used for solving conflicts between norms in case of incoherence (Kollingbaum *et al.*, 2006), when, for instance, an agent could be constrained by two contradictory norms. *tc* specifies when the norm is valid: before ('<'), while ('='), or after ('>') a date.

Obligations of the responsibility model are composed of a deontic operator (Obligation must of time) and a task (for instance, 'Must retrieve the log' is composed of the operator 'Must' and the task 'retrieve the log'). In order to perform a parallel between the responsibility model and the normative organisation model of MAS, each task composing an Obligation of the responsibility model is transformed into a root goal of the FS. These root goals are then decomposed in sub-goals coming from the capabilities needed to achieve the Obligation and gathering in mission (*m*). Rights needed to achieve Obligations are seen as condition (φ) making the corresponding norm (*norm*) been valid. To summarise, a Responsibility and its linked Obligations is seen as a role and all norms where it is the *bearer*. Figure 2 illustrates main elements of a simplified view of *M*OISE^{Inst} (in blue) pointing their matches in a detailed view of responsibility model in green.

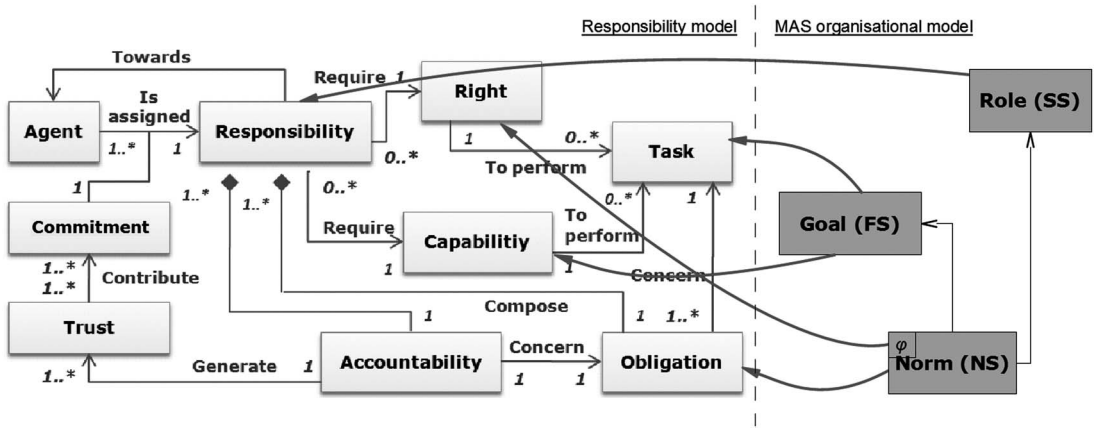

Figure 2 Match between responsibility model and normative organisation model

3.2 Assurance and trust

Although many conditions may need to be fulfilled to ensure that an agent meets its responsibilities, it is imperative that the following ones are met:

1. Rights: the set of rights entrusted to the agent must enable the satisfaction of the agent's obligations.
2. Capability: the overall capability assigned to an agent must be below the agent's intrinsic capability. Moreover, such capability should enable the agent to fulfil its obligations.
3. Level of trust: should be higher or equal to a minimum (predefined) threshold ($T_{\min}$).

Based on the above requirements the assurance for an agent fulfilling the obligation should be based on: *Assurance for fulfilment of obligation 'o' by an agent with right 'R', capability 'C', and trust 'T_P': $A_o(R, C, T_P)$.*

No assurance:

$$A_o\left(R, C, T_p\right) = 0 \; if \; (R_o \notin R) \cup (C_o \notin C) \cup \left(T_p \geq T_{\min}\right) \tag{1}$$

Otherwise:

$$A_o\left(R, C, T_p\right) = 1 \tag{2}$$

where R is the current rights of the agent; C the current capabilities of the agent; R_o the set of rights necessary for fulfilling obligation o; C_o the set of capabilities necessary for fulfilling obligation o; T_p the trust at period p.

Relations (1) and (2) imply that the satisfaction of an obligation can only be guaranteed if the set of rights allocated to the agent and the current capabilities are both subsets of the set of rights and capabilities required for the satisfaction of the obligations and if the trust level at period (T_p) is higher or at least equal to the reference $T_{\min}$. It is noteworthy to mention that more than one agent may fulfil such requirements and subsequently, the decision to select one of those as an alternative to a faulty agent will be based on the highest level of T_p.

From the MAS point of view the main concern is how to develop an organisation infrastructure that ensures the satisfaction of the organisational constraints and norms (e.g. agents playing the right roles, committing to the allowed missions). Many implementations of the organisation infrastructure follow the general architecture depicted in Figure 3. Domain agents are responsible to achieve organisational goals and use an *organisational proxy* component to interact with the organisation. The *organisational layer* is responsible to bind all agents in a coherent system and provides some services for them (Kitio *et al.*, 2007). In particular, they are in charge of verifying above conditions in order to decide of a potential reorganisation. The level of trust T_p of each of the component is provided by the organisational layer based on direct information as detailed in Guemkam *et al.* (2011).

The next section describes a case study that we adopt to illustrate the reassignment of an obligation based on assurance and trust values.

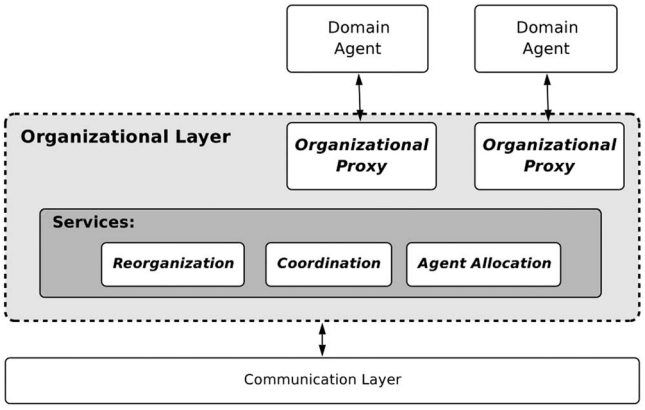

Figure 3 Common organisation implementation architecture for open multi-agent systems (Kitio *et al.*, 2007)

4 The broadcasting mechanism case study

The broadcasting mechanism (as depicted in Figure 4) aims at sending alerts to the population using media, such as Short Message Service (SMS) whenever a severe weather alert occurs. For that, sensors are disseminated on three layers corresponding to geographical areas (city, region, or country) and retrieve information pertaining to pressure, temperature, and electric voltage from probes located within a weather station and from the electrical grid. Regarding the different layers, sensors and aggregators have specific responsibilities:

- The Alert Correlation Engine (ACE) collects, aggregates, and analyses weather information from the probes deployed over the network and weather stations.
- Confirmed alerts are sent to the Policy Instantiation Engine (PIE). The PIE receives confirmed alert from the ACE, sets the severity level, and the extent of the geographical response. The PIE also instantiates high-level alert messages to be deployed.
- Finally, the high-level alert messages are transferred to the Message Supervising Point (MSP).

5 Assigning the responsibilities to the agents

5.1 Implementation

The architecture is composed of different types of agents that play a collaborative role. The agent architecture presented in Figure 4 is based on the ReD's one (Gâteau *et al.*, 2009). It proposed a solution to enhance the detection/reaction process and to improve the overall resilience of critical infrastructures. We extended the aforementioned architecture to accommodate the need for the responsibilities reassignment. The main agents involved include the (a) ACE, (b) PIE, (c) MSP, and (d) Message Broadcasting Point (MBP).

The MSP (Figure 5) is composed of two modules: (a) the policy analysis (PA) and (b) the Component Configuration Mapper. The PA is in charge of analysing the policies previously instantiated by the PIE. For that, the Policy Status database stores all of the communication policies and the corresponding status (in progress, not applicable, bypassed, enforced, removed) so that the PA module can check the consistency of the newly received message to be deployed. The Component Configuration Mapper module selects the appropriate communication channel.

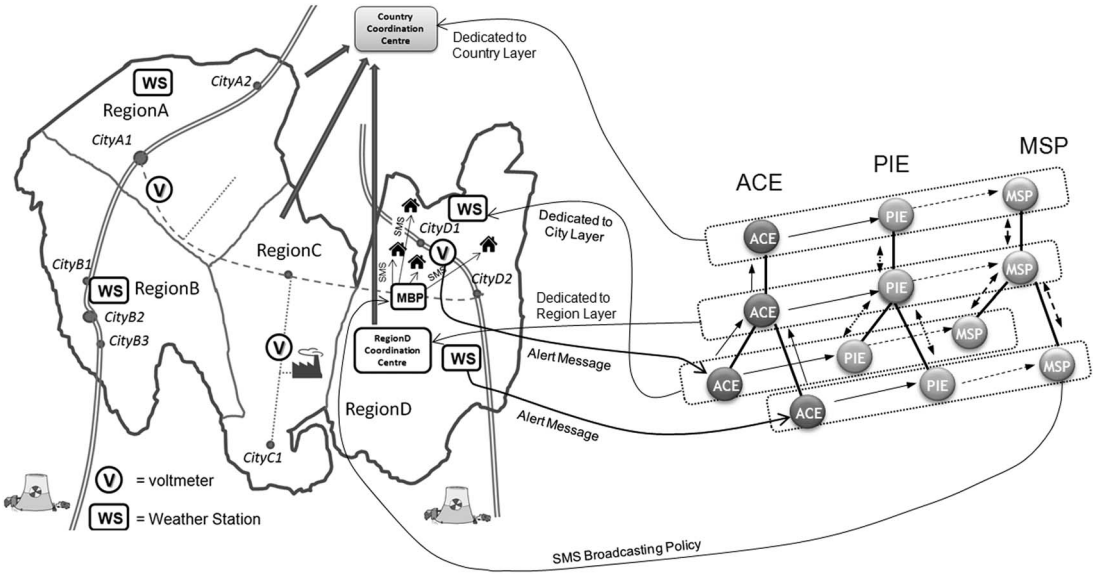

Figure 4 Broadcasting mechanism of weather alerts

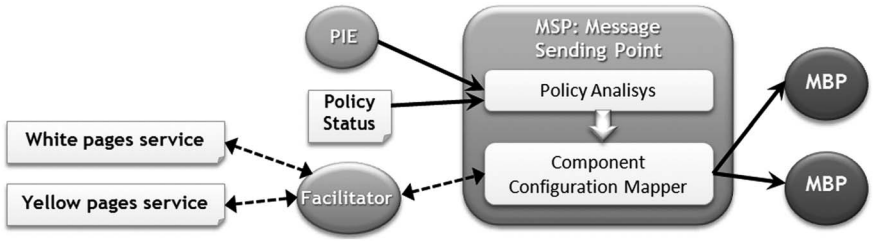

Figure 5 Message Supervising Point (MSP) architecture

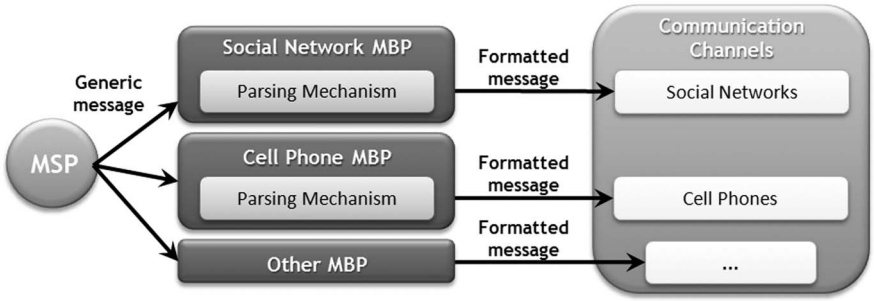

Figure 6 Message Broadcasting Point (MBP) architecture

The MBP is in charge of receiving the generic alert messages from the MSP. Then a specific parser converts the incoming alert message to the appropriate format according to the channel. Figure 6 presents two different kinds of MBPs. Different communication channels (e.g. SMS, e-mail, micro-blogging) to send alerts to citizens, hospitals, etc. are used. Consequently, our electric blackout prevention system is easily extensible for future communications facilities.

To consider the centralised trust of each agent, the organisational layer instantiated here with $\mathcal{U}$TOPIA from Schmitt *et al.* (2011) maintains a database of levels of trust of agents regarding the status of the accomplishment of their missions. This means that the MBP, the ACE, and the MSP have dedicated levels of trust. $\mathcal{U}$TOPIA's context permits to assign and manage the responsibilities of the components, such as MBP and MSP. Thereby, $\mathcal{U}$TOPIA controls the assignment of roles in an intelligent manner by taking into account the current level of assurance of a given norm to be respected as well as the trust level T_p of the agents.

Figure 7 introduces the extended ReD architecture illustrated with the weather broadcast alert system. The flow begins with an alert detected by a probe (voltmeter and weather station). The alert is sent by the MBP to the ACE agent (city layer) that confirms or not the alert to the PIE. Afterwards, the PIE decides to apply new policies or to forward the alert to an ACE from a higher layer (region layer). The PIE agent sends the policies to the MSP agent that decides which MBP is able to transform the high-level alert message into an understandable format for the selected communication channel (Feltus *et al.*, 2010).

To manage access rights, we add to the architecture a context rights management (CRM) module (right-hand side of Figure 7). The CRM is linked to the database storing information on the agents' rights which allows it to provide and amend the access rights to agents. It is also linked to the Context Manager of $\mathcal{U}$TOPIA to detect changes in the context, particularly when an agent has not sufficient capabilities anymore, as described in Guemkam *et al.* (2011).

5.2 Organisation

Based on the agents' responsibility model shown in Figure 2, we define the responsibilities of each agent within the architecture. Table 1 summarises the necessary capabilities and the rights for the agents playing the MBP role to accomplish a given obligation. We observe that the responsibilities include obligations, such as the obligation O_1 to retrieve the logs from the component it monitors and O_2 to provide an

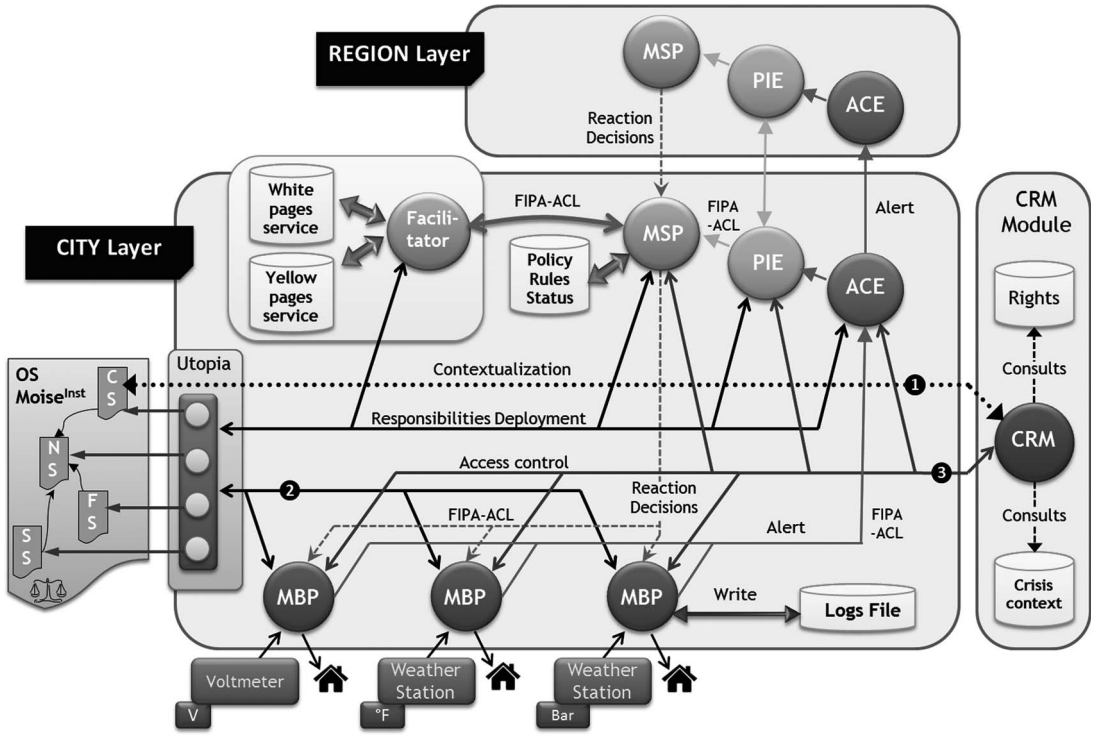

Figure 7 Detailed reaction architecture for power distribution adaptation based on weather parameters

immediate reaction if necessary. To perform the latter obligation O_2, it must have the capabilities to be on the same network as the component it controls, such as voltmeter, thermometer, or barometer (C_1), to be able to communicate with the MSP (C_2), with the facilitator agent (C_3), etc. It also must have the rights R_1 to read the log file on the concerned network component, R_2 to write the log in a central logs database, and finally R_3 to be able to read the Policy in the MAS management layer.

5.3 Reorganisation

In the considered scenario, we assume that at time t, the GSM/GPRS network is down, which impacts the rights and capabilities of the agents to fulfil their respective obligations. Table 2 provides the new capabilities and rights of the agent MBP-SMS as well as the corresponding assurance values to meet a given obligation. Such assurance value is based on the metrics provided in Section 3. After taking into account the specifications of the responsibilities associated with each MBP agent (cf. Table 1), one can assess whether current rights, capabilities, and trust level of MBP agents can be sufficient to fulfil a given obligation.

Let us consider, for instance, the information of MBP-SMS in Table 2. The current status of that MBP is such that it will not be able to fulfil the obligation O_2. The obligation to provide an immediate reaction is hindered by the fact that the MBP does not have the capability to communicate with the MSP (C_2). This means that any appropriate policy cannot be grounded to the MBP and be implemented in case of abnormality within the infrastructure. In the event that the assurance value for meeting a given obligation is '0', like in the cases discussed above, such an obligation is dedicated to an agent having the required rights and capabilities and belonging to the same group. For that, the following steps are executed (labelled from 1 to 3 in Figure 7):

1. The CRM inspects the contextual state of the institution and detects that MBP-E-MAIL has the capability to take the MBP-SMS's responsibility. In order to transfer the responsibility from MBP-SMS to MBP-E-MAIL, the CRM simply sends the suitable transitions to $\mathcal{U}$TOPIA in order to change the contexts (more information about contexts and responsibilities in Guemkam *et al.*, 2011).

Table 1 Message Broadcasting Point responsibilities specifications

Capabilities and rights	Agent's obligations	Mapping of capabilities to obligations	Mapping of rights to obligations
Capabilities			
C_1: is on the same network as the component to control	O_1: must retrieve the logs from the component it monitors	C_1, C_4, C_6, C_7	R_1, R_2, R_4
C_2: be able to communicate with the MSP			
C_3: be able to communicate with the facilitator agent			
C_4: have enough computing resource to monitor the component to control	O_2: must provide an immediate reaction if necessary	C_1, C_2, C_3	R_1, R_2, R_3
C_5: be able to communicate with the MAS management layer			
C_6: must be able to encrypt data	O_3: must communicate with the facilitator in order to get the address of the other components (MSP, ACE)	C_3	
C_7: be able to communicate securely with the ACE			
Rights			
R_1: allow to read log file on the concerned network component			
R_2: allow to write log in the central logs database	O_4: must report the incident to the ACE in a secure way	C_5, C_6, C_7	R_5
R_3: be able to read the policy in the MAS management layer			
R_4: allow to read and write in the alert database			

ACE = Alert Correlation Engine; MAS = multi-agent systems; MBP = Message Broadcasting Point; MSP = Message Supervising Point.

Table 2 Rights and capabilities of MBP-SMS at time *t*

Obligations concerning tasks	Trust level	Current agents' capabilities	Current agents' obligations	Assurance of obligation fulfilment
O_1: must retrieve the logs from the component it monitors	$T_p = 0.5$	C_1, C_4, C_6, C_7	R_1, R_2, R_4	1
O_2: must provide an immediate reaction if necessary		C_1, C_3	R_3	0
O_3: must communicate with the facilitator in order to get the address of the other components (MSP, ACE)		C_3		1
O_4: must report the incident to the ACE in a secure way		C_5, C_6, C_7	R_5	1

ACE = Alert Correlation Engine; MAS = multi-agent systems; MBP = Message Broadcasting Point; SMS = Short Message Service.

2. This allows $\mathcal{U}$TOPIA to reorganise the agents' responsibilities taking into consideration their obligation fulfilment assurance (i.e. having rights and capabilities to accomplish the missions and having a trust level higher than the T_p needed). As a consequence, MBP-SMS looses its access rights, is isolated, and its responsibilities are transferred to the MBP-E-MAIL.
3. The MBP-E-MAIL agent that receives the additional MBP-SMS's responsibilities requests the underlying access rights to the CRM.

This top-down approach based on an *a priori* OS is centralised (one $\mathcal{U}$TOPIA instantiating one organisation linked to one CRM with centralised databases), however, agents are autonomous (they are supervised by $\mathcal{U}$TOPIA but not controlled) and able to choose which goal to reach. In this use-case, some agents are only used as sensors (e.g. MBP) and thus do not achieve their mission because they are able to do it. Goals of MBP agents are mainly to sense the distributed system and deploy some new responsibilities. We could imagine other use-cases in which agents would be able to respect or not their obligation because of local objectives to reach. Responsibilities are considered as security policies and are translated and interpreted in *M*OISE^{Inst}'s norms. *M*OISE^{Inst} is an OML, not a security policy language. Our MAS system framework based on ReD architecture is able to take into account several standard security message formats such as XACML or OrBAC for access control, CIM-SPL for policy language, and even IDMEF for intrusion detection. In this work, we add to the framework the possibility to interpret responsibilities of the model and evaluate the assurance of obligation fulfilment. Assurance evaluation and trust-based reorganisation is a solution to improve security. If an agent becomes malicious, it will no more respect its obligations and as a consequence its responsibilities will be transferred.

6 Conclusion

This paper presented a novel approach to address the dynamic reassignment of an agent's responsibilities to its peer when it becomes unable to carry out its obligations. Indeed, MAS are widely used for the monitoring of distributed systems, but the assignment of the agents' functions is undertaken before the MAS system deployment. The proposed model exploits the concepts of assurance and trust as the indicators for identifying, respectively, when an agent becomes unable to meet its obligations, and for selecting the alternative peer that is believed to have a higher reliability for carrying out the task. An instantiation of the model has been presented on a weather station case study. The architecture is developed using ReD and $\mathcal{U}$TOPIA and has been shown to be effective during our first simulations, in ensuring the continuity of the monitoring in the event an agent was to lose some of its capabilities or rights. We simulated several responsibility assignment and reassignment for validating the approach and monitored the behaviours of the MBP, MSP, and ACE deployed in different location with different parameters. However, the instances discussed in this paper only considered one faulty agent at a time.

Our future work will therefore be directed towards the consideration of more complex scenarios, such as agents failing simultaneously and consecutively after the occurrence of an adverse event. The comparison of

our solution to existing ones will also be conducted. Its performance will be measured using projects such as BARWAN, which focuses on enabling truly useful mobile networking across an extremely wide variety of real-world networks and mobile devices. We plan also to adapt our architecture in the context of cloud computing. The goal will be also to detect defective agent and reorganise them in order to make them able to respect negotiated and signed service-level agreement between a service provider and a service consumer.

Acknowledgement

This work was partially funded by TITAN Project (C08/IS/21), financed by the National Research Fund of Luxembourg.

References

Abielmona, R. S., Petriu, E., Harb, M. & Wesolkowski, S. 2011. Mission-driven robotic intelligent sensor agents for territorial security. *IEEE Computational Intelligence Magazine* **6**(1), 55–67.

Baig, Z. A. 2012. Multi-agent systems for protecting critical infrastructures: a survey. *Journal of Network and Computer Applications* **35**(3), 1151–1161.

Boissier, O. & Gâteau, B. 2007. Normative multi-agent organisations: modeling, support and control. In *Proceedings of the Dagstuhl Seminar 07122: Normative Multi-agent Systems*. Boella, G., v. d. Torre, L. & Verhagen, H. (eds.). Vol. II. ISSN 1862-4405.

Boudaoud, K., Labiod, H., Boutaba, R. & Guessoum, Z. 2000. Network security management with intelligent agents. In *Proceedings of the Network Operations and Management Symposium (NOMS 2000)*. IEEE/IFIP, 579–592.

Coutinho, L., Sichman, J. & Boissier, O. 2007. Organisational modeling dimensions for multi-agent systems. In *Proceedings of the Workshop Agent Organizations: Models and Simulations (AOMS@IJCAI07)*. Dignum, V., Dignum, F. & Matson, E. (eds).

Demazeau, Y. 1995. From interactions to collective behaviour in agent-based systems. In *Proceedings of the 1st European Conference on Cognitive Science*, 117–132.

Esteva, M., Padget, J. & Sierra, C. 2002. Formalizing a language for institutions and norms. In *Proceedings of the 8th International Workshop on Intelligent Agents VIII*, LNAI **2333**, 348–366. Springer.

Feltus, C., Khadraoui, D. & Aubert, J. 2010. A security decision-reaction architecture for heterogeneous distributed network. In *Proceedings of the Fifth International Conference on Availability, Reliability and Security ('ARES 2010 – The International Dependability Conference')*. IEEE.

Feltus, C. & Petit, M. 2009. Building a responsibility model including accountability, capability and commitment. In *Proceedings of the Fourth International Conference on Availability, Reliability and Security ('ARES 2009 – The International Dependability Conference')*. IEEE.

Ferber, J. & Gutknecht, O. 1998. A meta-model for the analysis and design of organizations in multi-agent systems. In *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS 1998)*. IEEE, 128–135.

Ferber, J., Michel, F. & Baez, J. 2004. AGRE: integrating environments with organisations. In *Environments for Multi-Agent Systems*. LNCS **3374**, 48–56. Springer.

Gâteau, B. 2007. *Modélisation et Supervision d'Institutions Multi-Agents*. PhD thesis, Ecole Nationale Supérieure des Mines de Saint-Etienne.

Gâteau, B., Boissier, O., Khadraoui, D. & Dubois, E. 2005. *M*OISE^{Inst}: an organizational model for specifying rights and duties of autonomous agents. In *1st International Workshop on Coordination and Organisation (CoOrg 2005)* affiliated with the 7th International Conference on Coordination Models and Languages, Namur Belgium. V. d. Torre, L. & Boella, G. (eds).

Gâteau, B., Boissier, O., Khadraoui, D. & Dubois, E. 2007. Controlling an interactive game with a multi-agent based normative organisational model. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II*. LNCS **4386**, 86–100. Springer.

Gâteau, B., Khadraoui, D. & Feltus, C. 2009. Multi-agents system service based platform in telecommunication security incident reaction. In *Proceedings of the Global Information Infrastructure Symposium (GIIS 2009)*. IEEE, 1–6.

Guemkam, G., Feltus, C., Schmitt, P., Bonhomme, C., Khadraoui, D. & Guessoum, Z. 2011. Reputation based dynamic responsibility to agent assignment for critical infrastructure. In *Proceedings of the 2011 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, **2**. IEEE, 272–275.

Hübner, J., Sichman, J. & Boissier, O. 2002. A model for the structural, functional, and deontic specification of organizations in multiagent systems. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence (SBIA 2002)*. LNAI **2507**, 118–128. Springer.

Jennings, N. R. 1999. Agent-oriented software engineering. In *Proceedings of the 9th European Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW 1999)*, LNCS **1647**, 1–7. Springer.

Kitio, R., Boissier, O., Hübner, J. F. & Ricci, A. 2007. Organisational artifacts and agents for open multi-agent organisations: 'giving the power back to the agents'. In *Proceedings of the 2007 International Conference on Coordination, Organizations, Institutions, and Norms in Agent Systems III (Coin 2007)*, 171–186.

Kolaczek, G. & Juszczyszyn, K. 2007. Traffic and attack pattern analysis for multiagent distributed intrusion detection system. In *Proceedings of the Intelligent Systems and Knowledge Engineering (ISKE 2007)*. Atlantis Press, 733–739.

Kollingbaum, M. J., Norman, T. J., Preece, A. & Sleeman, D. 2006. Norm refinement – informing the re-negotiation of contracts. In *Coordination, Organizations, Institutions, and Norms in Agent Systems II, LNCS 4386*. Noriega, P., Vázquez-Salceda, J., Boella, G., Boissier, O., Dignum, V., Fornara, N. & Matson, E (eds). Springer, 245–258.

Lesser, V. K., Decker, W. T., Carver, N., Garvey, A., Horling, B., Neiman, D., Podorozhny, R., NagendraPrasad, M., Raja, A., Vincent, R., Xuan, P. & Zhang, X. 2004. Evolution of the GPGP/TAEMS domain-independent coordination framework. *Autonomous Agents and Multi-Agent Systems* **9**(1), 87–143.

Li, W. & Hoang, D. 2009. A new security scheme for e-health system. In *Proceedings of the International Symposium on Collaborative Technologies and Systems*, 361–366.

Ouedraogo, M., Khadraoui, D., De Remont, B., Dubois, E. & Mouratidis, H. 2008. Deployment of a security assurance monitoring framework for telecommunication service infrastructure on a voip service. In *Proceedings of the New Technologies, Mobility and Security Conference (NTMS '08)*. IEEE, 1–5.

Pham, N., Baud, L., Bellot, P. & Riguidel, M. 2008. A near real-time system for security assurance assessment. In *Proceedings of the 3rd International Conference on Internet Monitoring and Protection*. IEEE, 152–160.

Pynadath, D. & Tambe, M. 2003. An automated teamwork infrastructure for heterogeneous software agents and humans. *Autonomous Agents and Multi-Agent Systems* **7**(1–2), 71–100.

Schmitt, P., Bonhomme, C., Aubert, J. & Gâteau, B. 2011. Programming electronic institutions with utopia. *Information Systems Evolution* **72**, 122–135.

Servin, A. & Kudenko, D. 2008. Multi-agent reinforcement learning for intrusion detection. In *Adaptive Agents and Multi-Agent Systems III*. LNCS **4865**, 211–223. Springer.

Singh, M. P. 2008. Semantical considerations on dialectical and practical commitments. In *Proceedings of the 23rd National Conference on Artificial Intelligence (AAAI)*, vol. 1. AAAI Press, 176–181.

Sommerville, I. 2007. Models for responsibility assignment. In *Responsibility and Dependable Systems*. Dewsbury, G. & Dobson, J. (eds). Springer, 165–186.

Stahl, B. 2006. Accountability and reflective responsibility in information systems. In *The Information Society: Emerging Landscapes*. Zielinski, C., Duquenoy, P. & Kimppa, K. (eds). Springer, 51–68.

Wooldridge, M. 2002. *An Introduction to Multi-Agent Systems*. John Wiley & Sons.